DISTRIBUTED SWITCH FABRIC ARBITRATION

Inventors: Guy M. Chemla

Peter Si-Sheng Wang

BACKGROUND OF THE INVENTION

Field of the Invention

[0001]   The present invention relates to high-capacity switch architectures for communication systems, and more particularly to scalable switch architectures with distributed arbitration logic.

Description of Related Art

[0002]   High-capacity communication switch architectures have been developed to address the growing numbers of users and uses of communication networks. The switch architectures are based on a variety of switch fabric designs, which includes a shared media switch in which either memory or bus resources are shared using time division multiplexing, a stacked Banyan, a mesh switch, a crossbar switch and others. There are various advantages and disadvantages of each type.

[0003]   In the implementation of high-capacity communication switches, the establishment of connections between ports on the switches is made in response to the traffic flowing through the switch. The connections are made in switching cycles that allow for efficient use of the resources. In each switching cycle of some embodiments, an arbitration process is executed by which competition for the use of the ports on the switches is resolved. As the number of ports on a given switch increases, the complexity of arbitrating among the ports increases dramatically and requires an equivalent increase in processing power. Such arbitration must be executed efficiently and quickly so that the performance of the switch is maintained. However, the computation of optimal connection maps within the switch among a large number of possible routes can be a difficult problem in dynamically changing conditions.

[0004]   It is often desirable to add capacity to a given switching system. The mesh type switch and the crossbar type switch are both capable of high-capacity switching, and have extendable architectures. The crossbar switches are believed to be suited for more efficient extensions in size by adding additional crossbar planes to a stack of crossbar planes, than are

- 1 -

mesh type architectures. So for the purpose of understanding the present invention, the basic components of a prior art crossbar architecture are shown in Fig. 1.

[0005] A generalized crossbar as shown in Fig. 1 includes a plurality of satellites S#1 through S#s, where the satellites correspond to network ports, router elements, line cards or other interface structures between communication networks and the switch fabric. The satellites communicate through a plurality of crossbars X-1 through X-m. The satellites communicate with the crossbars through respective sets of satellite to crossbar S2X links 1-1 to 1-x for satellite S#1, and s-1 though s-y for satellite S#s. The aggregate bandwidth for the satellite S#1 is expressed as the summation of the bandwidth of each of the satellite to crossbar links. The number of links from a satellite to a crossbar is dependent upon a particular implementation of the satellite, and may include one or more links per crossbar plane. The satellites include a plurality of links L to the communication channels external to the switch. In the example shown, the satellite S#1 includes n links within aggregate bandwidth equal to the summation of the bandwidth of each of the n links. The input bandwidth from the communication channel over the links L need not be equal to the bandwidth between the satellite and a crossbar over the links S2X, for buffered satellites.

[0006] A central arbitration entity 10 is coupled with the crossbars and communicates with each of the satellites through a control communication channel. The control communication channel 11 may be an inband channel which steals cycles from the crossbar switch, or any other type of communication media. A multiple plane crossbar switch, like that shown in Fig. 1, may support static grouping of the fabric pipes, where a group is formed by logically joining links between satellites and a crossbar. In this case, a transmission across a group must request a plurality of elements through the crossbar in order to make a single connection, with a wider bandwidth because of the multiple paths. Thus, for example, if the switch is configured into groups of two crossbar to satellite links, the switch operates at twice the bandwidth with half the port count. In any event, the generalized crossbar architecture of Fig. 1 supports a wide variety of bandwidth and port count configurations. However, the central arbitration entity limits the scalability of the structure because of the increasing complexity encountered as the number of satellite to crossbar links is increased.

[0007] With this background, and an understanding of the need for an improved arbitration algorithms for complex switches, it can be understood that a need exists for improved arbitration structures for complex communication switches.

-2-

SUMMARY OF THE INVENTION

[0008]   The present invention improves the scalability and throughput for high-capacity communication switches by providing a distributed arbitration algorithm which addresses the arbitration inefficiency of prior art systems. The invention is based upon distributed arbitration logic units in each satellite unit in a communication switch. The arbitration logic units execute a protocol for computing a connection map for the switch fabric during each arbitration cycle. The protocol includes a plurality of phases including broadcasting backpressure parameters among the arbitration logic units, generating bids for access to switch fabric resources in each of the arbitration logic units utilizing information shared with other arbitration logic units, broadcasting the results of the bidding process, and configuring the switch fabric using the results.

[0009]   Thus, one embodiment comprises a system for distributed control of a communication switch, in which switch satellites maintain ingress queues for inbound communications from the external communication channels to the switch fabric and egress queues for outbound communications from the switch fabric to the external communication channels. The combination of an ingress queue on a particular satellite and an egress queue on the same or another satellite establishes a virtual output queue. The system includes a plurality of arbitration logic units coupled with respective switch satellites. The arbitration logic units in the plurality include logic to control an arbitration cycle for a given switch cycle.

[0010]   The arbitration cycle includes a first stage in which performance parameters, such as backpressure parameters, are gathered from other arbitration logic units in the plurality. The performance parameters indicate a status of one or more egress queues maintained in the switch satellites coupled with respective arbitration logic units. A second stage of the arbitration cycle occurs in which bid data are propagated among the plurality of arbitration logic units. The bid data includes a set of bids for use of egress queues during the switch cycle. The bids in the set include a destination identifier indicating a destination egress queue in one of the plurality of switch satellites and a weighted pressure parameter indicating a result of the combination of the performance parameter of the destination egress queues with a condition such as forward pressure of the source ingress queue. The third stage of the arbitration cycle occurs in which a connection map based upon the bidding is computed. In one embodiment, the connection map is broadcast to all switch satellites involved in the arbitration cycle. In a fourth stage, the switch fabric is configured based upon the connection map.

[0011]   In another embodiment, the ingress and egress queues are configured as virtual output queues, in which the virtual output queues have priorities. In the case of queues with priorities, the weighted pressure parameter is a function of the priority of the virtual output queues involved. The priority of a virtual output queue may be established with reference to the egress queue priority, the ingress queue priority, or an independently established priority.

[0012]   In another embodiment, the plurality of arbitration logic units respectively include configuration logic that indicates a bid order. In the second stage of the arbitration cycle, a first arbitration logic unit in the order sends bid data to the next arbitration logic unit in the order. The next arbitration logic unit in the order consolidates and sends the bid data to a next arbitration logic unit, and so on until the last arbitration logic unit in the order receives the consolidated bid data. The bid data in the respective arbitration logic units is based upon the gathered performance data and the condition of virtual output queues associated with the inbound queues maintained in the respective switch satellites and any previous arbitration logic unit or units in the order. In one embodiment executed according to the bid order, the connection map is computed in the last arbitration logic unit in order, and broadcast to the other arbitration logic units in the system.

[0013]   In one preferred embodiment, the switch fabric comprises a crossbar switch with one or more planes. Further, in one embodiment, communication control logic is coupled with the switch fabric and supports synchronous communication among the plurality of arbitration logic units and the switch fabric for purposes of the arbitration cycles.

[0014]   In one aspect of the invention, a method is provided for distributed control of a communication switch, in which the communication switch comprises a switch fabric and a plurality of switch satellites. The method comprises steps including:

> gathering performance parameters in each switch satellite in the plurality from other switch satellites in the plurality, the performance parameters indicating a status of the one or more egress queues maintained in the respective switch satellites,
>
> sharing bid data among switch satellites in the plurality, the bid data including a set of bids for use of egress queues during the switch cycle, the bids in the set including a destination identifier indicating a destination egress queue in one of the plurality of switch satellites, and a pressure parameter indicating a result of a combination of the performance parameter of the destination egress queue with a condition of a source ingress queue,

computing a connection map based on the bidding in at least one of the switch

satellites in the plurality, and

configuring the switch fabric based upon the connection map.

[0015] In one embodiment of the method, the step of indicating a bid order is included. In this embodiment, the sharing of bid data includes sending bid data in the order from a first switch satellite to a next switch satellite until the last switch satellite in the order receives the bid data. The bid data at the respective switch satellites is based upon the gathered performance data and a condition of inbound queues associated with particular virtual output queues that are maintained in the respective switch satellites and any previous switch satellite or satellites in the order.

[0016] In another embodiment, the method includes:

gathering performance parameters in each switch satellite in the plurality from other

switch satellites in the plurality, the performance parameters indicating a

backpressure of the one or more egress queues maintained in the respective

switch satellites;

sharing a bid data matrix among switch satellites in the plurality, the bid data matrix

including a set of bids including one bid for each of the virtual output queues for

use of egress queues during the switch cycle, the bids in the set including a

pressure parameter indicating a result of a combination of the performance

parameter of the destination egress queue with a condition of the ingress queue of

the virtual output queue;

computing a connection map based on the bidding in at least one of the switch

satellites in the plurality; and

transmitting a vector to the crossbar switch for configuring the crossbar switch based

upon the connection map.

[0017] The bid data matrix in one embodiment comprises a data structure for holding a bid for each of the virtual output queues serviced by the crossbar switch. The process includes indicating a bid order of the switch satellites. The step of sharing the bid data matrix includes computing a bid data matrix in a first switch satellite having entries for virtual output queues which originate in the first switch satellite, sending the bid data matrix to the next switch satellite in the order where the next switch satellite re-computes the bid data matrix with entries for virtual output queues originating in the first switch satellite and in said next switch satellite,

and sending the bid data matrix to the next switch satellite and so on, until the last switch satellite in the order receives consolidated bid data and finally computes the bid data matrix with entries for virtual output queues originating in all the switch satellites.

[0018]   By removing the processing load of the arbitration process from the switch fabric, the present invention allows building of switches with varying capacity using simple building blocks so the higher capacity products resemble in architecture and behavior products of lower capacity. Furthermore, switches according to the invention can be expanded to increase performance over time with improved technology.

[0019]   Other aspects and advantages of the present invention can be seen on review of the figures, the detailed description and the claims which follow.

BRIEF DESCRIPTION OF THE FIGURES

[0020]   Fig. 1 is a schematic diagram of a generalized crossbar switch architecture according to the prior art.

[0021]   Fig. 2 is a simplified diagram of a communication switch architecture with distributed arbitration logic according to the present invention.

[0022]   Fig. 3 is a diagram illustrating data flow for a protocol for distributed arbitration in the communication switch of Fig. 2 according to the present invention.

DETAILED DESCRIPTION

[0023]   A detailed description of an embodiment of the present invention is provided with respect to Figs. 2 and 3, in which Fig. 2 shows a communication switch improved with the distributed arbitration logic units of the present invention. Fig. 3 illustrates a representative protocol used for computing a connection map during an arbitration cycle according to the present invention.

[0024]   The switch architecture of Fig. 2 includes a switch fabric 100 such as a crossbar switch fabric including a plurality of crossbar planes. A plurality of satellites 101, 102, 103 is coupled with the switch fabric 100. The satellites 101, 102, 103 maintain respective ingress Virtual Output Queues VOQs and egress queues. Thus, satellite S1 includes ingress queues VOQ1-1 to VOQ1-6, satellite S2 includes ingress queues VOQ2-1 to VOQ2-6, and satellite S3 includes ingress queues VOQ3-1 to VOQ3-6. Likewise, each of the satellites 101 through 103 includes a plurality of output (egress) queues. Satellite S1 includes output queues OQ1 and

OQ2 , satellite S2 includes output queues OQ3 and OQ4, and satellite S3 includes output queues OQ5 and OQ6. Each of the satellites includes links to external communication networks. Thus, satellite S1 includes the set of links 104, satellite S2 includes the set of links 105 and satellite S3 includes the set of links 106. According to the present invention, arbitration logic units 101, 102 and 103 are associated with respective satellites S1, S2 and S3. In addition, control communication controller 107 is coupled with the switch fabric 100 to support the arbitration protocol. Each of the switch satellites includes a plurality of satellite to crossbar links 110, 111, 112. Typically, there is at least one link per plane in the crossbar. Where each plane serves a number X of ports for connection to satellite to crossbar links, and each crossbar has the number X of planes, there would typically be the number X links between each satellite S1, S2, S3 and the switch fabric 100. A control channel for use during the arbitration cycle is also included as indicated by lines 115, 116 and 117 between each of the satellites S1, S2, S3, respectively, and the control communication controller 107.

[0025]   The control communication controller 107 is in charge of forwarding control messages such as backpressure parameters and final arbitration decisions among the arbitration logic units. There is typically at least one control communication controller located on each crossbar plane. Preferably the control communication controller is a low latency device such that processing by it has no significant effect on the speed of operation. The control links 115, 116, 117 connecting each satellite to the control communication controller are connected to arbitration logic units 101, 102, 103 in each of the satellites S1, S2, S3. The control links 115, 116, 117 provide channels by which the arbitration logic units S1, S2, S3 communicate in order to share, update and compute in a distributed manner, the arbitration results for the next transfer cycle through the switch fabric.

[0026]   The arbitration process is synchronous, and a start cycle signal is applied at the beginning of every arbitration cycle to all system components in a preferred embodiment. Upon initialization, the components are configured with all the parameters required to operate, including a relative position of each satellite, a number of virtual output queues and so on. Because of the synchronous nature of the preferred embodiment, the control communication controllers and the satellites can use the start signal and parameters in order to deduce the precise sequencing of each phase and sub-phase of the arbitration protocol.

[0027]   The arbitration protocol can be understood with reference to both Figs. 2 and 3. In a preferred embodiment, the arbitration cycles can be considered in four phases:

1. Broadcast the backpressure;

2. Bid;

3. Broadcast results;

4. Configure.

[0028]   Each of the four phases is considered in sequence. During the first phase, each satellite having an output queue sends backpressure status for its queues to all other satellites in the system via the control links and the control communication controllers. In Fig. 3, this phase is referred to as the gather backpressure parameters phase, in which the matrix 200 is shown. The numbers in the first column correspond to the satellite S1 with arbitration logic unit 101, the numbers in the second column correspond to satellite S2 with arbitration logic unit 102, and the numbers in the third column correspond to satellite S3 with arbitration logic unit 103. Thus, the first column indicates that the output queue 1 with priority 1 (1,1), output queue 1 with priority 2 (1,2) and so on. In Fig. 3, the numbers computed by each satellite arbitration logic unit is printed in a different font.

[0029]   The backpressure parameters from the first satellite S1 comprise the first column in this manner because of the presence of the output queues 1 and 2 in the first satellite S1. Likewise, the second column represents the backpressure parameters for output queues 3 and 4 with priorities 1 and 2. The third column represents the backpressure parameters for the third satellite with output queues 5 and 6 having priorities 1 and 2. As can be appreciated, as the number of output queues and priorities increase, the size of the matrix increases dramatically.

[0030]   In this manner, each arbitration logic unit in the system gathers the matrix 200. The control communication controllers receive the data and broadcast the matrix to all satellites, preferably simultaneously or essentially so, forwarding columns as they come. At a first time, all satellites should have the backpressure matrix 200 for all egress queues. The data items are transmitted in order so their identification is implicit in a preferred embodiment. The size of each data item is based on the number of possible values the backpressure can take. For example, a four-valued backpressure indicating levels such as empty, fairly empty, fairly full and full, requires two bits.

[0031]   While receiving the backpressure matrix 200 from its neighbors, and using the forward pressure for virtual output queues for each destination and priority, each satellite computes a weighted pressure. Computation of the weighted pressure is accomplished by first computing the differential pressure (dp) for each destination, which is a function of the forward

- 8 -

pressure of the ingress queue VOQ on the particular satellite to which the switched traffic is directed, and a backward pressure for the output (egress) queue of the destination satellite to which the channel is directed. Then, according to the priority of the output queue, the differential pressure is converted to the weighted pressure. The exact function used for

5   computation of the differential pressure and of the weighted pressure based upon the forward pressure, the backward pressure and priority is implementation dependent. Other status and performance conditions of the queues and the satellites can be utilized in the computation of the weighted pressure.

[0032]   Using a bid process, the distributed arbitration scheme collects forwarding threads of

10   the source to the destination queues according to the various priorities. The result of this process is a legal connection map. The process causes the highest pressured queues to bubble up as requests are passed through the bidding satellites.

[0033]   The process will also try to maximize the number of switched units per arbitration cycle. Note that in the particular case of cell switching, where the switched unit is a self-

15   contained datagram, there is no reason to limit a given destination to one source per priority queue. Packets are usually switched in several switching cycles, so transferring more than one switched unit to the same queue/priority using many links may mix the parts of independent packets irrecoverably.

[0034]   The first satellite S1 with ingress queues computes its needs for all queues of the first

20   destination satellite, and places the result in the first column of matrix 201. While repeating the same for the next destination, (second column of matrix 201), at time t2 in Fig. 3, satellite S1 sends the four components of the first column (two output queues times two priorities) through the control links, to the second satellite S2.

[0035]   Each component of the vector has the form (source S, wp), source S is one of

25   satellites S1, S2 or S3, and wp is the weighted pressure. Note that the matrices 201, 202, 203 of Fig. 3 list the item worked upon, not the transmitted result. All elements departing the first satellite S1 would have the form (1, wp): 1 being the name for satellite S1, and wp the weighted pressure for a given queue. The data items are transmitted in order and are implicitly identifiable by the link and the time. The coding of the source can advantageously use incremental length

30   coding; this will be explained later.

[0036]   At time t4, the first satellite S1 would have finished sending its matrix 201 to satellite S2. Starting at time t2, when it has received the data for the first column of matrix 201, the

second satellite S2 will compute its needs, make its bid in the form of matrix 202, and send the results to the third satellite S3, where beginning at time t3 or thereafter, matrix 203 is computed.

[0037]   The bid work of the first satellite S1 is quite simple. For the next and following satellites, there is more to do:

           1) Compare own wp for a given queue to that received from the previous satellite;

           2) Update the matrix with the results; and

           3) Send the matrix to the next satellite.

[0038]   None of the participants is allowed to overload any destination, that is in the preferred system, none is allowed to request the transfer to a destination, of more switched units than is physically achievable in one cycle. If more queues have a positive wp than the available load, the highest should be elected. The cases of equal wp should be resolved using Round Robin or randomization scheme.

[0039]   Bid fairness can be assured following an algorithm such as the following. Let R be a matrix within the S element, of p lines by w columns whose elements are random numbers ranging from 1 to the number of the S element; e.g., R1 of S1 is not used. R2 of S2 has random numbers ranging from 1 to 2 and R3 of S3 has random numbers ranging from 1 to 3. All R's components are updated every arbitration cycle (not every bid transfer).

[0040]   Let,

$dp-(q, p)$, be the received dp for queue q, priority p

$dp0(q, p)$, be the self dp for queue q, priority p

$dp+(q, p)$, be the transmitted dp for queue q, priority p

source, be the transmitted bid winner for this queue / priority

[0041]   If $dp-(q, p) \neq dp0(q, p)$, then the highest value is sent as $dp+(q, p)$ and the source is its owner.

[0042]   If $dp-(q, p) = dp0(q, p)$, then its owner is elected as source for only one value of the corresponding random number $R(q, p)$, e.g., for the value 1. This strategy insures fairness of the bubbled-up winner even in the case of all equal bids.

[0043]   Example on S3

Possible values for R3 are 1 or 2 or 3

If $R3(q, p)=1$, then $dp+(q, p) = dp0(q, p)$: the random being 1, S3 wins

If $R3(q, p)=2$, then $dp+(q, p) = dp-(q, p)$: the random being 2, S3 loses

If $R3(q, p)=3$, then $dp+(q, p) = dp-(q, p)$: the random being 3, S3 loses

[0044] Sharing the destination bandwidth is accomplished as follows in one example. If a destination is fully loaded, then, in order to satisfy a satellite's own higher wp bid, it should purge a lower wp in order to make room. In case of equal-valued lower wp, deletion should use randomization or Round Robin. This is the reason why, in order to make to bid for a queue in a destination, the satellite element needs to know the status of all requests for that destination. In the following example, in the notation in each entry in the matrix is a three integer value: q.p:u, indicating that the wp = u for absolute queue q (numbered 1..6), priority p.

[0045] Assume satellite S1 has following needs, indicated by the entries in the matrix with non-zero weighted pressure.

$$
\begin{array}{lll}
1.1:0 & 3.1:0 & 5.1:0 \\
1.2:0 & 3.2:0 & 5.2:0 \\
2.1:0 & 4.1:0 & \mathit{6.1:3} \\
2.2:0 & 4.2:0 & \mathit{6.2:2}
\end{array}
$$

and satellite S2 has following needs

$$
\begin{array}{lll}
1.1:0 & 3.1:0 & \mathit{5.1:4} \\
1.2:0 & 3.2:0 & 5.2:0 \\
2.1:0 & 4.1:0 & 6.1:0 \\
2.2:0 & 4.2:0 & 6.2:0
\end{array}
$$

Further assume that there are just two logical links through the switch fabric, so all three bids cannot be satisfied. Then, satellite S2 should delete satellite S1's bid for queue.priority 6.2 and elect satellite S2's bid for queue.priority 5.1.

[0046] Source overload is supported by the arbitration logic in one preferred embodiment, and is handled in one example as follows. A satellite S element can make requests in excess of its own load (dequeue capacity). In the case of Fig. 2 with four crossbar planes, for example, a source can request to dequeue more then 4 queues (while respecting the rule of avoiding destination overload). The condition is that the excess (e.g. above 4), in the order of the destination scan -- hence chronological order -- has a lower wp than the first four. Successor satellite S element is allowed to win artificially over these excess requests until its own load capacity is satisfied. Thereafter, bidding can continue as in the normal case on the source

overloaded capacity. The last satellite S is not allowed to make source overload and must delete all excess requests from all sources, which it does not need to use.

[0047]   In the following examples, the notation q.p:u means wp = u for absolute queue q (numbered 1..6), priority p:

5

Case 1) Assume satellite S1 has following needs:

1.1:0	*3.1:3*	5.1:0
1.2:0	*3.2:4*	5.2:0
2.1:0	*4.1:3*	*6.1:2*
2.2:0	*4.2:2*	6.2:0

10

Satellite S1 could make all these requests.

Case 2) If the need for 6.1 were 6.1:4, it would need to limit it requests to

1.1:0	*3.1:3*	5.1:0
1.2:0	*3.2:4*	5.2:0
2.1:0	*4.1:3*	*6.1:4*
2.2:0	4.2:0	6.2:0

15

Case 1.1) Assume satellite S1 has needs of case 1

1.1:0	*3.1:3*	5.1:0
1.2:0	*3.2:4*	5.2:0
2.1:0	*4.1:3*	*6.1:2*
2.2:0	*4.2:2*	6.2:0

20

25   and satellite S2 has following needs

1.1:1	3.1:0	5.1:0
1.2:1	3.2:0	*5.2:1*
2.1:0	4.1:0	*6.1:1*
2.2:0	4.2:0	6.2:0

30

Then satellite S2 would artificially win for 6.1 though its wp is 1 whereas S1's is 2.

[0048]   Un-biasing the destinations can be addressed by the following process.   The process of avoiding source overload may result in biasing of destinations due to the sequential scan.

35   Biasing can be avoided by randomizing the order of the list (columns of matrices 201, 202, 203 in Fig. 3). Synchronization can be assured by starting with the same seed and using the same pseudo-random generator on every arbitration entity of every S.

- 12 -

[0049] Avoiding starvation in queues can be addressed as follows. A scheduling based only on the differential pressure dp, may induce indefinite starvation to a low dp queue. This can be avoided by requesting each S to artificially inflate the dp of a low dp queue which has not been served for a while.

5

[0050] Preserving the packet's order can be addressed as follows. Once the first transfer unit of a packet has been transferred to a given destination priority queue, all consecutive units should be transferred uninterrupted by the same source. This can be achieved by changing the dp to the highest reserved value indicating 'locked' for this destination, until the packet end.

[0051] Hot insertion of new satellites which results in a form of synchronized random

10 sequence, can be dealt with either by self-synchronizing sequences or by resetting all random generators at the beginning of the cycle in which the new satellite S participates. Likewise, there should be means to update the relative position of the satellite S elements.

[0052] The case of overflow in any queue is handled, preferably, by dropping packets at source as opposed to destination in order to spare the fabric bandwidth. In case of overflow at

15 the input of one transfer unit, it may be interesting to flush the remainder of the packet.

[0053] Incremental length source coding may be used. The length of the source field transmitted in the bid phase does not need to be constant. One can see that satellite S1 does not need to send a field identifying itself at all as whatever reaches satellite S2 comes necessarily from satellite S1 in the order. Likewise, source transmitted from satellite S2 to satellite S3 is

20 either 1 or 2 and can be coded with one bit. Generally, the source field length can be coded with the number of bits required to code the (S number −1) or indeed the S number if these were counted from 0.

[0054] The bid process is repeated by all three satellites. At t3, the third satellite S3 broadcasts its bid result, which should be legal, i.e. free of source or destination overload, via the

25 control communication controllers. This should be used as a connection map to itself and all neighboring satellites. Once the map transmitted to all satellites, each destination can proceed to the configuration of the crossbars.

[0055] In the final phase of the arbitration cycle, the crossbars are configured. Each destination satellite, using its control links, sends at t5 simultaneously to the attached crossbars

30 one scalar, identifying the source it wants to dequeue using the corresponding data communication link. Both the source and the destination would know, at this point, which source / priority to dequeue and which destination / priority to enqueue, using that link.

[0056]   The process described above should be regarded as one example of implementation. Alternative processes can trade complexity against performance. For example, a high bandwidth control communication controller and associated links together with higher processing power of the arbitration entities within the satellite S elements can be used. In such an implementation, many columns -- ultimately all columns -- of the matrices can be transmitted at once for a very fast arbitration cycle.

[0057]   Other algorithms can be deployed so as to speed up the bid process, e.g., to be proportional to the logarithm of number of satellites instead of the number of satellites. Indeed, many distributed sorting algorithms can be applied to the bid process.

[0058]   While the present invention is disclosed by reference to the preferred embodiments and examples detailed above, it is to be understood that these examples are intended in an illustrative rather than in a limiting sense. It is contemplated that modifications and combinations will readily occur to those skilled in the art, which modifications and combinations will be within the spirit of the invention and the scope of the following claims.